Aurrerapenak semi-infinito programetan:Aplikazioak eta algoritmoak

Zer dira semi-infinito programak? Programa hauek azken batean pisu handia duten exekutagarriak baino ez dira, hainbat para metro erabiltzen dituztenak bere metodoak erabili ahal izateko, ondorioz programa hauek ganbilak direla esan ohi da, izan ere, programa bere loop ia infinitura heldu baino lehen hainbat aldagaien atzipena egiten du eta loop-aren ondoren aldagai hauen balioak returneatzen ditu.

Historian zehar infiniturainoko optimazioari buruz hitz egitean Chebyshev aproximazion pentsatu egiten dugu

$$CA: \quad \min_{\boldsymbol{p}} \|F(\cdot) - a(\boldsymbol{p}, \cdot)\|_{\infty, Y} \quad \text{s.t.} \quad \boldsymbol{p} \in P.$$

$$\begin{aligned} SIP_{CA}: \quad \min_{\boldsymbol{p}, q} q \quad \text{s.t.} \quad & F(\boldsymbol{y}) - a(\boldsymbol{p}, \boldsymbol{y}) \leq q, \ \boldsymbol{y} \in Y, \\ & -F(\boldsymbol{y}) + a(\boldsymbol{p}, \boldsymbol{y}) \leq q, \ \boldsymbol{y} \in Y, \\ & \boldsymbol{p} \in P, \end{aligned}$$

Hala ere galdetuko duzue, non daude aurrerapen hauek, aurrerapen hauek infiniturainoko optimizaziorekin lantzean erabiltzen diren metodoetan daude:

Diskretizazioa: Metodo hau erabiltzean orain ez gaude lehen bezala P azpimultzo ia infinitu bat erabiliz, baizik eta, M azpimultzo finitu bat erabiliz. Hau erabiltzean, programa handia atal txikiagoetan banatzea lortu egiten dugu. Hala ere, konkurrentziari dagokioenez metodo hau ez da optimoena, izan ere, prozesadore batek ezin du M nahi azpimultzo eratu eta hauek prozesatu

$$M_{LBP}(Y_k) = \{\boldsymbol{x} \in \mathbb{R}^n | \ g(\boldsymbol{x}, \boldsymbol{y}) \leq 0 \text{ for all } \boldsymbol{y} \in Y_k\} \supseteq M.$$

Gehiegi estimatutako metodoa: Kasu honetan M azpimultzo finitu baina oso handia erabili ordez M azpimultzoan dauden parametro esanguratsuenen analisia bakarrik egiten da modu honetan optimizazioa ezin hobea da, baina igarpena txikitu egiten da.

Azkenik esan beharra dago, aplikazio hauek “lower-level convexity”-an eragina dutela gehienbat, baina aldi berean programa handiak konkurrentzia handiko ordenagailuetan/zerbitzarietan exekutatzerakoan, duten zama murriztea lortzen dela.